AS

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/822,300	03/23/2001	Georg Muenzel	00 P 7515 US 01	8037

7590 01/26/2005

Siemens Corporation
Intellectual Property Department
186 Wood Avenue South
Iselin, NJ 08830

EXAMINER	
VU, TUAN A	

ART UNIT	PAPER NUMBER
2124	

DATE MAILED: 01/26/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 09/822,300	**Applicant(s)** MUENZEL, GEORG
	Examiner Tuan A Vu	**Art Unit** 2124

-- The MAILING DATE of this communication appears on the cover sheet with th correspond nce address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 03 September 2004.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-52 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-52 is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) □ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

DETAILED ACTION

1. This action is responsive to the application filed 9/3/2005.

Claim 45 has been amended and claims 1-52 have been re-submitted for examination.

Claim Rejections - 35 USC § 101

2. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

3. Claim 36 is rejected under 35 U.S.C. 101 because the claims are directed to a non-statutory subject matter.

The Federal Circuit has recently applied the practical application test in determining whether the claimed subject matter is statutory under 35 U.S.C. § 101. The practical application test requires that a " useful, concrete, and tangible result" be accomplished. An "abstract idea" when practically applied is eligible for a patent. As a consequence, an invention, which is eligible for patenting under 35 U.S.C. § 101, is in the "useful arts" when it is a machine, manufacture, process or composition of matter, which produces a concrete, tangible, and useful result. The test for practical application is thus to determine whether the claimed invention produces a "useful, concrete and tangible result".

As per claim 36, the claim only recites a storage medium having stored thereon a control code representation formatted in a markup language; and does not recite any action step for implementing what is recited as control code. The claim only provide descriptive elements without specifying actions performed by those elements; hence fails to provide steps leading to a useful, concrete, and tangible result as required by the practical application test. Hence, the claim only amounts to an abstract idea without a practical and useful purposes, hence is rejected for leading to a non-statutory subject matter.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1-5, 7-10, 12-13, 15-17, 19-23, 25-28, 30-32, 34-44, and 46-52 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lau, USPN: 6,598,219 (hereinafter Lau), in view of Hoskins et al., USPN: 6,167,406 (hereinafter Hoskins).

As per claim 1, Lau discloses a method for representing a object and task-oriented computer program code using a graphical programming language tool that stores the created code in the computer memory in an internal representation during execution, the method comprising:

identifying the created code in computer memory in the internal representation (e.g. *storage device* - col. 2, line 66 to col. 3, line 14; Fig. 1-2 – Note: tree structure for storing design objects teaches inherent storage of objects to be identified for interface display and editing); and

converting the code from the internal representation to a markup format (e.g. col. 3, line 21 to col. 4, line 19; col. 6, line 46 to col. 18, line 59).

But Lau does not disclose that the task-oriented program code is an industrial automation computer program code; however, Lau discloses that the program code is for business application encompassing a task oriented structure and model (col. 1, lines 26-57; col. 2, line 66 to col. 3, line 12). The use of models to represent real-world entities via an interfacing tool for building a business application program in a variety of domains of industry was a known concept at the time the invention was made. One such domain can vary from business related applications to manufacturing design or industrial automation control. In a method using a object-oriented modeling tool analogous to Lau, Hoskins discloses using browser technologies and markup language, e.g. SGML and activeX, to transport application program development and related representation across platforms and to facilitate developers builder environment (e.g. col. 11, lines 50-63; col. 12, lines 47-65) and further discloses a framework to implement automation control using editing interface to implement a ladder logic in relation to a Programmable Logic Controller to effect the controlling tasks(col. 12, line 66 to col. 13, line 51; Fig. 2-80). It would have been obvious for one of ordinary skill in the art at the time the invention was made to apply the object-oriented modeling framework and markup conversion as taught by Lau in the field of industrial automation control as taught by Hoskins because like other business applications, the industrial domain application can also benefit from the readily available internet and/or browser technologies, e.g. SGML transport protocol and Java language extensions, so that in conjunction with the modeling tool as taught by Lau, task and object-oriented oriented applications in the business sector as well as in the manufacturing industry can also be implemented in one cost-efficient and extended manner.

As per claim 2, Lau discloses storage of the markup-formatted code (e.g. col. 2, lines 35-50; col. 3, lines 41-64; col. 21, lines 23-42 – Note: data in extensible form (XML) for import and export and being displayed for editing discloses inherent storage for transport across the internet).

As per claim 3, refer to Lau: col. 2, lines 35-50.

As per claims 4 and 5, Lau discloses converting the markup-formatted code to the internal representation in computer memory (e.g. *parsed and rendered* -- col. 21, lines 11-23) and representing such code in a corresponding graphic format (*object builder* - col. 21, lines 11-23; Fig. 1; col. 6, lines 22-34 – Note: hierarchy created by the modeling tool and converted into DTD/XML format implicitly disclose re-conversion back into such hierarchy of design components).

As per claim 7, Lau discloses XML (e.g. col. 3, lines 14-40).

As per claims 8, and 10, Lau disclose modeling (Fig. 1-2); hence has if not explicitly from the Figures shown, then implicitly disclosed graphical language comprising a flowchart, block diagram, and sequential diagram.

As per claim 9, Lau does not teach graphical programming language comprising a ladder logic, but in view of the teachings by Hoskins, providing a ladder logic to be implemented in XML form for transmission would also have been obvious for the same rationale as set forth in claim 1 above.

As per claims 12, and 15, refer to claims 8, and 10 for corresponding rationale.

As per claim 13, see claim 9.

As per claim 16, Lau discloses an editing interface to enable the user to perform creating a file and define the objects (e.g. Fig.1-2; col. 5, lines 27-45; col. 6, line 16-10); hence has disclosed editor and generating screen objects which trigger inherent commands to generate of metadata in terms of XML or DTD formatted files.

As per claim 17, see Lau (col. 6, lines 22-34; col. 21, lines 11-42).

As per claim 19, this is a computer product with computer-readable medium (see Lau: col. 21, lines 55-62) for performing the same steps limitations recited respectively in claim 1;

hence is rejected with the corresponding rejections as set forth in claim 1, including the rationale to address the industrial automation computer program code limitation.

As per claims 20-23, refer to the rejections of claims 2, 4, 3, 5, respectively.

As per claims 25-28, refer to claims 7-10, respectively.

As per claims 30-32, refer to claims 12, 13, 15, respectively.

As per claims 34-35, refer to claims 17 and 16, respectively.

As per claim 36, Lau discloses storage of program design representation code formatted in markup language (e.g. e.g. col. 3, line 21 to col. 4, line 19; col. 6, line 46 to col. 18, line 59; Fig. 1-2); but does not disclose industrial automation control code. But this limitation has been addressed in claim 1 above.

As per claim 37, see claim 7.

As per claim 38, Lau implicitly discloses coupling to remote computer system (e.g. *import…export* - col. 2, lines 30-50 – Note: using markup language implicitly disclose transporting across some network in order to be rendered at the receiving end compatible with SGML and related protocols).

As per claim 39, Lau discloses a computer program product for permitting a user to create software programming control code, comprising a computer-readable storage medium (col. 21, lines 55-62) having a computer program code on it, the code comprising:

graphical programming language code, an editor adapted to permit the user to create the programming control code using graphical elements (e.g. Fig.1-2; col. 5, lines 27-45; col. 6, line 16-10), the code being stored in an internal representation during execution (col. 2, line 66 to col. 3, line 14); and code for converting the graphical programming code thus stored from the internal representation to markup language format (e.g. col. 3, line 21 to col. 4, line 19; col. 6, line 46 to col. 18, line 59).

But Lau does not disclose that the task and object-oriented programming control code is an industrial automation computer program code; but this limitation has been addressed in claim 1 above.

As per claim 40, Lau discloses converting industrial automation control code from the markup language format to the internal representation (see rejection of claim 4).

As per claim 41, Lau discloses a method for communicating the logical structure of software programming control data in order to permit a plurality of application developers to create applications relating to the data, the method comprising:

creating a schema defining a content model for markup language files generated by the programming control program system (e.g. col. 3, line 21 to col. 4, line 19; col. 6, line 46 to col. 18, line 59); and

posting the schema for access over the network by the application developers (e.g. *import...export* - col. 2, lines 30-50; Fig. 1-2 – Note: user interface for editing and rendering data into and from XML for modeling data is equivalent schema made available for access to plurality of developers when markup development/modeling content are transferred from machines to machines).

But Lau does not disclose that the software programming control data is industrial automation control data; but this limitation has been addressed in claim 1 above.

As per claims 42 and 43, refer to claim 7-8, respectively.

As per claim 44, Lau discloses a method for providing software programming control code from a system of developers coupled in a network (*import...export* - col. 2, lines 30-50 – Note: using markup language implicitly disclose transporting across some network in order to be rendered at the receiving end compatible with SGML and related protocols), the system comprising:

accessing a markup-formatted version of the control code (col. 2, line 66 to col. 3, line 14; col. 3, line 21 to col. 4, line 19);

transmitting the accessed markup-formatted control code over the network, thereby causing the markup-formatted control code to be received by the receiving system (e.g. *parsed and rendered* -- col. 21, lines 11-23).

But Lau does not explicitly disclose a server system coupled over a network to a client system; and the transmitting the control code over the network in connection with a client system network address, such client system being the receiving system. The use of SGML format and XML format by Lau is strongly suggestive that network protocol is being used and system being coupled together are inherently connected by network address. Hence, the client-server limitation is implicitly disclosed.

Nor does Lau disclose that the software programming control code is industrial automation control data; but this limitation has been addressed in claim 1 above.

As per claim 46, Lau discloses modeling to support a business application programming scheme using a modeling tool (re claim 44) but fails to disclose using mail message for communications. Official notice is taken that in an enterprise wherein multiple users are connected via the enterprise network services such that network communication and data distribution help fulfill the enterprise business applications, the use electronic mail to communicate data was a well-known concept at the time the invention was made. The providing of electronic mail to Lau's system so as to enable multiple developers to communicate with the common framework to retrieve markup-formatted control data would have been obvious in light of the benefits related to such type of communications as suggested by the well-known concept from above.

As per claim 47 and 48, see Lau (col. 2, lines 31-65).

As per claim 49, this claim includes an obvious variation of claim 44, and is rejected using the rationale set forth in claim 44 to address the transmitting of control data based on the network address of the first client system.

As per claim 50, this claim includes the same limitation of claim 4 or 40; and is rejected with the rationale used in claim 4 or 40 in conjunction with the rejection as set forth in claim 49; because in a network where markup data is distributed, rendering such data back into internal representation by a first, a second or a third client would be the same.

As per claim 51, Lau discloses a method for programming software control code, comprising:

providing a computer system coupled to a network (e.g. col. 2, lines 31-50 – Note: Markup language and SGML implicitly discloses network being used for coupling systems together);

configuring a first computer to receive over the network transmissions of data from a plurality of software developer systems (*import...export* - col. 2, lines 30-50; Fig. 1-2 – Note: user interface for editing and rendering data into and from XML for modeling data is equivalent schema made available for access to plurality of developers when markup development/modeling content are transferred from machines to machines); and

receiving data from a the plurality of programming control code developer systems program code in a markup language format (e.g. col. 3, line 21 to col. 4, line 19; col. 6, line 46 to col. 18, line 59).

But Lau fails to disclose that the software programming control code is industrial automation control data. This limitation is rejected herein using the same rationale as set forth in claim 1 above.

As per claim 52, see claim 7.

6. Claims 6, 11, 14, 18, 24, 29, 33, and 45 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lau, USPN: 6,598,219 (hereinafter Lau), in view of Hoskins et al., USPN: 6,167,406 (hereinafter Hoskins), as applied to claims 1, 17, 19, 25, 44; and further in view of Suzuki et al., "Making UML models exchangeable over the Internet with XML: UXF approach", 1998 (hereinafter Suzuki).

As per claim 6, Lau does not explicitly disclose converting the markup-formatted code to the internal representation with a browser. But the fact of converting program design data into XML form implicitly discloses re-conversion from markup language back to the internal representation at the device that retrieves the markup-formatted data, using browser like document rendering utilities. Suzuki, in a method to convert modeling language specification or metadata analogous to Lau's into XML formatted data, discloses API for reconverting the markup format back into the data structures stored in repository (pg. 7, section 4.4). In case Lau does not specifically teaches such converting back to the internal representation in computer memory, it would have been obvious for one of ordinary skill in the art at the time the invention was made to provide the browser application programs as taught by Suzuki to convert markup formatted design representation data back into internal representation because the very purpose of having data converted into markup format is to facilitate transport using internet protocol so that the original data can be reconverted for use from browser utilities that can parse and process markup formatted data as intended by both Lau and Suzuki.

As per claims 11 and 14, Lau discloses modeling (Fig. 1-2) with implicitly teaching of graphical language comprising a flowchart, block diagram, but does not explicitly disclose sequential diagram. In view of Suzuki's method of using UML in conjunction with the mark-up conversion approach (*sequence diagram* - see sections 1, 2, 3, 4.1,4.2, 4.3, pg. 1-6), it would have been obvious to provide a modeling wherein the graphical programming language would include sequential diagram as taught by Suzuki to Lau's method because the time-dependent aspect of a interrelated activities in a program application would be enhanced by such type of diagrams in order to carry the task-oriented applications as intended by Lau or Hoskins.

As per claim 18, Lau discloses displaying code (e.g. *data objects, flowmark, methods, source* - Fig. 1- 2); but does not expressly disclose browser; but the use of browser has been addressed in claim 6 above.

As per claims 24, 29, and 33, refer to the rejections of claims 6, 11, and 14, respectively.

As per claim 45, Lau discloses transmitting the markup formatted control code and causing said control code to be received at the client system (refer to rejection of claim 44). Lau, however does not disclose that the client device transmit to the server system data relating to the automation to which the markup formatting is directed; and that in response to receipt from the client system, the server system has access to the modified control code; and causes to transmitting of the modified control code to be received at the client system. Hoskins, teaches enterprise-wide distribution of control data using a central modeling system (e.g. Fig. 2, col. 12, line 66 to col. 14, line 56) and Suzuki, providing a client-server paradigm wherein the server stores and provide modeling data to client via the protocols for providing markup format for interconnected developers to model their respective their target application (see sections 1, 2, 3, 4.1,4.2, 4.3, 4.4, pg. 1-6- emphasis on chap. 4). It would have been obvious for one of ordinary skill in the art at the time the invention was made to provide the ability for a server to store modified markup control data in response to request by client and thereby distribute such control code back to the client including the modified data according to the client specifications; and apply that to Lau's method so that the server is the system to store modified control data/code and effect to providing of such control code to requesting client system as suggested by Hoskins and Suzuki. One of ordinary skill in the art would be motivated to do so because the purpose of modeling is to be able to accommodate the implementation of as many applications as possible and the fact of converting into a markup format is to enable multiple developing system to access the control data, and distributing control data so as to fulfill the change requests from remote clients would enhance the applicability of the multi-users modeling purposes of Lau's product in light of Suzuki's and Hoskins suggestions, i.e. a wide-spread concept that is to use a central service system to provide upgraded version of control code or software to requesting clients in response to specifications provided by such request.

Response to Arguments

7. Applicant's arguments filed 9/3/2004 have been fully considered but they are not persuasive. Following are the corresponding Examiner's rebut in regard thereto.

Rejection 35 USC 101:

(A) Applicant has submitted that claim 36 is rejected under 35 USC 101 as lack of utility; and that the invention is in compliance with 35 USC 112, first paragraph (Appl. Rmrks, pg. 11, bottom, pg. 12-13). The rejection clearly states that the invention lacks action leading to a concrete, tangible and useful result as required by the Practical Application Test, hence is leading to a non-statutory subject matter. There is no rejection under lack of utility. There are provisions in the MPEP about what constitutes statutory subject matter and what does not, particularly in the computer related field; and Applicant is asked to refer to § 2106, chp. IV: "Determine whether the claimed invention complies with 35 U.S.C. 101".

Thus, Applicant's argument about lack of utility is moot or misapplied.

(B) Applicant has submitted that the Examiner should specifically explain the scientific basis for his or her factual conclusions (Appl. Rmrks, pg. 13, bottom, pg 14) to establish prima facie as to why an invention lacks utility. Again, the rejection is about a non-statutory type of rejection under 35 USC 101; and the rejection has pointed out that claim 36 by reciting just a 'representation of industrial control code formatted in a markup language' does not amount to making the claimed invention statutory, i.e. with step actions yielding interaction of recited elements to accomplish a result, such result being required to be concrete, tangible and useful in the computer art, as per the Practical Application Test. Thus, the arguments are not persuasive and/or misplaced, i.e. they appear to stem from the construing that a non-statutory rejection is the same as a lack of utility rejection.

(C) Applicant has asked that presentation of evidence be provided to support the use of 'official notice' in the Office Action (Appl. Rmrks, pg. 15, bottom). The prior art that have communicating network in association with a business/enterprise based type of development are for example: USPN 6,721,713, USPN: 6,640,249, USPN: 6,636,242, and USPN: 6,160,477; all of which teach notification of clients in the network via Fax or Email.

(D) Applicant has submitted that Hoskins teaches away from using a 'markup language format' (Appl. Rmrks, pg. 16, top 2 para) for reasons listed in Hoskins col. 12, li 4-12. The rejection does not address the limitation of making HTML as one implementing language in industrial automation but only as a communicating languistic means. The issue at stakes is not that Hoskins should provide both developing an industrial automation and implementing it with markup language. The rejection clearly states:

"In a method using a object-oriented modeling tool analogous to Lau, Hoskins discloses using browser technologies and markup language, e.g. SGML and ActiveX, to transport application program development and related representation across platforms and to facilitate developers builder environment (e.g. col. 11, lines 50-63; col. 12, lines 47-65) and further discloses a framework to implement automation control using editing interface to implement a ladder logic in relation to a Programmable Logic Controller to effect the controlling tasks(col. 12, line 66 to col. 13, line 51; Fig. 2-80)."

The means for transporting specifications for developing a system like an industrial automation is what the rejection is alluding to. Such means is via a browser based platform in which markup language is inherent and via which objects can be tagged or embedded with Java extensions such as ActiveX functions in this case. The missing feature by Lau is that the markup representation, which is a common language standard in network communication and enterprise development tool, is not applied in the field of industrial automation -- working on using a ladder logic for example. The rationale is to make the obvious connection between such specification language means of communicating and a application domain that Lau does not particularly disclose. That is why Hoskins is brought in so to provide such field of application (i.e. industrial automation) while at the same time using a markup language to convey application program specifications. Even though Hoskins mentions about drawbacks on using strictly HTML as specification tool in a development, the bottom line is that Hoskins adds additional Java based implementation to reinforce the static browser pages so to render them more dynamic (see Hoskins: col. 12, lines 47-65); and by means of which convey the code specification, and support thereby a seemingly network-based system development functionalities. The fact that Hoskins applies embedded Java snippets or extensible objects inside tags of a browser pages (ActiveX) does not signify that only Java code implementation is the sole language that Hoskins would rely on to transport the development specifications leading to code implementation. The use of browser with dynamic Java code added thereto to transport such Java object embedded inside pages is just evidence that Hoskins does not do away with what is known as HTML tag based methodology (see *dynamic web pages*, *Web document*, *applet* - col. 12, lines 20-46). Besides, the use of markup language is not what is missing in the combination as set forth in the rejection. As intended, the rejection evolves around whether somebody has applied a markup specification in supporting industrial automation development at the time the invention was made. And Hoskins teaches both dynamic web pages and OLE/ActiveX. For the sake of clarifying as to why ActiveX is not teaching away from dynamic browser pages, the use of embedded objects as in OLE (via using Javascript and HREF tag) onto Java based browser methodologies was a well known concept; and in light of Hoskins' browser communicating means supported by (as opposed to completely substituted by) ActiveX, one of ordinary skill would recognize the very necessary interrelation between the Java based embedding of objects and dynamic development via browser HTML pages as clearly shown by Hoskins. Hence, the prima facie rationale as to why the use of markup language as communicating means to help convey specification for or to support the industrial automation development has been established for the reasons above.

(E) Applicant has submitted that the 'implicitly disclosed' phrase used by the Office Action in regard to claims 8, 10 and the 'trigger inherent commands' are just allegations for failing to establish a 'necessarily present' descriptive material required by a prima facie (Appl. Rmrks, pg. 16, bottom para).

First, for claim 1, Lau discloses flow chart and graph (Fig. 2); then there is no need to provide evidence.

Second, as for claim 16, when a user manipulates the model by reviewing the tree of elements and using interacting means (see col. 21, lines 12-37; Fig. 1) -- like a mouse click to modify the tree -- the presence of a command as to enable the markup specification to be generated or modified (e.g. in DTD or XML) is understood or recognized via such mouse event. The graphical interface (Fig. 1-2) enabling the user to reedit the structure of the hierarchical specification reads on what is claimed as editor. There is no explicit distinction in the claim that the so-called 'command' has to be strictly of one form and not another, like a console text command versus a graphical event based command. Hence, the user manipulating the tree display from Fig. 2 discloses what is recited by claim 16. In other words, by clicking on the tree structure, a command is sent to modify the markup file as displayed from col. 14-19 of Lau. The step of sending this editor command is hence necessarily present under the action taken by the user re-editing the graphical model. Because Applicant's argument is not persuasive, the rejections will stand as set forth.

Conclusion

8. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (272) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571)272-3719.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence – please consult Examiner before using) or 703-872-9306 (for official correspondence) or redirected to customer service at 571-272-3609.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

VAT
January 13, 2005

KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100